

Dynamic Resource Monitoring of SaaS with Attestation for a Trusted Cloud Environment

Udhayakumar Shanmugam¹ and Latha Tamilselvan²

¹Research Scholar, ²Professor,

^{1,2}School of Computer, Information and Mathematical Science,
B.S. Abdur Rahman University, Vandalur, Chennai 600048, Tamil Nadu, India
mailtoudhay@gmail.com, latha.tamilselvan94@gmail.com

Abstract

Network-centric applications are built on an intricate infrastructure that binds the communication services to distribute across heterogeneous environments. These services are becoming increasingly innovative and autonomic to execute on demand processes on a virtual platform. This kind of collaboration has fueled the growth of business for a landscape change, creating the era of cloud computing. Present security exploits in this uncharted landscape require a fabric of a trustworthy networked society, which must be inherently secure and reliable. Consequently, every resource that is shared in the cloud is not secure enough, hence, the need to monitor these shared services for its trustworthiness has aroused. In our work, we propose a trusted computing model that monitors services offered to a user through behavior analytics. An Application Monitoring Engine (AME) gets invoked at runtime to detect any changes in the behavior and analysis its pattern. Deviations if any, is escalated into a threat, which is verified through a centralized trust repository and subsequently attested by Behavior Analytics and Attestation Server (BAAS). Earlier models assess trust based on reputation, service agreements, fuzzy and probability, which are mostly static in nature and does not certify the trust value. Our model focuses trust in dynamic nature, and also certifies it through remote attestation protocol. The model is implemented in an OpenStack cloud setup for its feasibility and the performance is analyzed for an image editing software service.

Keywords: *Trusted computing, cloud security, software as a service, attestation, behavior analytics*

1. Introduction

Today, development of a complex, intelligent and networked information services forms the future needs of organizations. Business focus on this terrain has led the information technology sector to the development of virtualized and automated services widely known as cloud computing. The growth of cloud adoption can be attributed to the business community, whose primary motive is to reduce the capital expenditure (CAPEX) and operational expenditure (OPEX). The initial phase of cloud adoption has been a success story, but major security breaches and attacks have reduced their reliance and confidence towards the cloud. This is because the private and public implementation of cloud, are merely an amalgamation of heterogeneously developed computing paradigms working in tandem. Therefore, security implications and vulnerabilities that are in existence will still continue to pose threats to the environment. Confidence in cloud security is currently at an all-time low; with potential consumers remain hesitant, despite the rapid escalation of cloud usage. Hence leading service providers such as Amazon, Microsoft, and Google adopts stringent standards and implements state-of-the-art security measures to prevent any sort of vulnerability. Further, they routinely administer stress test and strengthen these measures through International security

standards like the ISO/IEC 27017 [1]. Notwithstanding these measures, there have been several security breaches and attacks, relentlessly made to gain access to critical infrastructure. If this is the case, with the market leaders of cloud, then small and medium enterprises (SMEs) would naturally have to rely on the public cloud as partners rather than develop a secure private infrastructure. Taking into consideration how greatly this would increase operational expenditure, it is inherently safe to have a private security perimeter to protect in-house data and lend non-critical data to the public cloud. So, in order to have a secure cloud setup we need to implement a multi-layered infrastructure to monitor and assess resources. The assessed services report back with a detailed measurement for predictive analytics. Over a period of time, the analytics will generate a list of potential causes that can have varying consequences. These predictions can also help in establishing a sustained and trusted relationship. Thus, trust becomes a key requirement for successful cloud service operations. The users will believe that their data is safe at the providers end, and the providers will do the best to reinforce this belief. Like assuring trusted service, protecting the privacy of user data form the core of security infrastructure. Hence setting the level of compliance with the cloud providers and agreeing upon certain Service Level Agreements (SLAs) can ensure that data is privacy-protected [2]. If any services in the cloud are assured with privacy preservation objective through the enforcement of behavioral policy then its stakeholders especially consumers, providers, vendors and brokers are likely to trust that environment.

The proposed work focuses on establishing trust through behavior monitoring, security considerations and privacy associations. Rather than attempting to judge the security of a system, our model tries to verify whether the security provided is certified as per the standard guidelines. It checks SLA and ensures that they are met without any deviations. This process is carried out by measuring the values generated during the process of applications life cycle. Further to the process of monitoring the behavior of resources, the resultant values are checked against well-known values that are stored in the centralized database. Once a decision has arrived based on the trust score, third-party attester takes charge of certification. Any cloud broker, who is associated with a recognized security infrastructure, is the attester who measures the authenticity of the measurements from the source. Overall, a trusted environment through remote attestation ensures the growth of cloud computing, which in turn builds confidence and enhances the reputation of a service.

The paper is organized as follows: It commences with a review of related work undertaken for resource monitoring, of cloud services in a trusted environment. The proposed methodology for evaluating behavioral pattern and their underlying causes has been explained with detailed architecture in Chapter 3. An adaptation algorithm for efficient trust categorization, with the phases required to make an attestation, is given in Chapter 4. Finally, in Chapter 5 procedures for implementation with a result analysis are provided. Chapter 6 concludes with future projections of trust and its migration into a virtual domain.

2. Related Work

Cloud security and trust models are two major issues in cloud computing paradigm. By far, most research works have been focused towards instituting trust, in the cloud through the standards framed by the Trusted Computing Group (TCG) [3]. TCG specifies key functionalities for IT security, specifically on cloud security through the concept of a Trusted Multi-tenant Infrastructure (TMI). The architecture of TMI is based on the development of a Trusted Platform Module (TPM) chip, which is a crypto processor embedded inside special processors meant for workstations and servers, released by Intel. It performs key encryptions and platform authentication that acts as roots of trust. With respect to the attestation mechanism, a Client-Oriented Remote Attestation (CORA)

model is proposed [4], for selecting a node in the cloud. It has various security levels, corresponding to their own particular needs, and dynamically verifies the node's security status. With respect to trust and its hierarchies, work has been carried out for the Internet of Things (IoT), where a hierarchical trust mechanism puts forward a verifiable caching interaction digest schema. The paper also analyzes the application's features and trust demand like the credibility of the detaching mechanism and reader trust [5]. An adaptive trust model for software services in a hybrid cloud environment where formal trust metrics - like the temporal nature of concurrent services and completion of service - decide the trusted nature of the software is proposed [6]. On trading privacy and trust in online interactions, the authors have categorized the processes of trading privacy for trust into symmetric privacy-for-trust negotiations and asymmetric privacy-for-trust negotiations. The asymmetric privacy is further dividing into privacy-revealing and privacy-preserving subcategories and evaluates privacy for trust gained [7].

To discuss how the concept of a secure and trusted environment can be applied to maintain the authenticity and integrity of digital evidence, a paper has been proposed that includes the unity of six components. They are forensics policy, security policy, model and trusted management system, trusted computing, secure channel communication, and the human factor [8]. Work related to classifying trust in two broader senses has been studied where in a socio-economic strand, trust needs to address standards and interoperable technology with the impact of policy and legal issues relative to cyber trust [9]. Non-technical issues such as ethics, sociology, culture, psychology and economy are other deciding factors that challenge a trustworthy system. Trust, Privacy, and Security (TPS) figure among the most prominent issues that have been in existence for years. New and evolved security problems - including identity theft, collusion attacks, side-channel attacks, cross-site scripting, phishing, and sabotage are multiplying rapidly with increased frequency and sophistication. Whereas, on the other hand, the number of recent publications, patches and intrusion detection from security experts offering viable solutions to offset threats to vulnerable security systems is decreasing. Thus, the need to improve the average consumer's confidence on the internet requires transforming the entire cloud platform into a trusted cloud environment.

3. Trust Establishment

Trust is a complicated phenomenon associated with miscellaneous disciplines and influenced by measurable and non-measurable factors [10]. Establishing trust in a digital environment especially in the cloud, where resources are scattered across the globe, involves processing virtual and physical measurements that may comprise real or faked identities. In this multi-tenant, resource pooling architecture, procuring accurate information and arriving at a dynamic analysis is always a complex task. However, if trust could be established through a computing model, then surely system security and personal privacy would be improved with a great deal. Trust in general can be defined as "a state involving confident positive expectations, about another's motives with respect to oneself in situations entailing risk" [11].

Trust and trusted computing with its security issues, have been the focus of the computing world for the last decade. A trusted computing infrastructure guarantees *control* of data to provide the transparency that can be verified by a customer. Trust in cloud computing requires data to be digitally signed for integrity and hence *privacy* preservation must be provided through efficient cryptographic techniques. Since data is physically spread across multiple data centers, a consumer would never know exactly where his data is; hence control over the data is minimal. Therefore, diminishing control and transparency are key factors to be addressed. In order to provide the user with full control, a remote access tool can be enabled on the consumer side that can disable or enable data manipulation commands. It is also imperative that policies that are enforced

be adhered by users so that the cached information can be deleted once the user has logged off. Further to the model of remote access control, it is vital that the cloud provider does not disclose all the data to another. Instead, only a part of a cloud or a *virtual* setup within the group or enterprise should be made visible. Thereby, ensuring that others data are not mixed up or visible to another. This eliminating cross-VM attack and side-channel attack [12]. To provide users *transparency*, it is necessary to build enterprise security architecture around cloud communication. Providing foolproof security is of prime importance, but we should also take care to judge the security capabilities of a system through third party *certification*. For a trusted service, it is vital to ascertain whether encryptions are foolproof and can withstand attacks. Thus, any service provided by a service provider must pass through a complete check by third-party experts. The experts assess and deliver a score that acts as a trust enabler. Finally, every cloud provider should highlight their success stories through feedback for *reputation* index. This index can help other potential users to identify and select products. Also, a novel method of providing security certification through star rating will certainly enhance the reputation of that service, and further enabling it to be recommended across other service providers. A chain of trust can be formed with similar providers through federated trust values for a global trustworthiness.

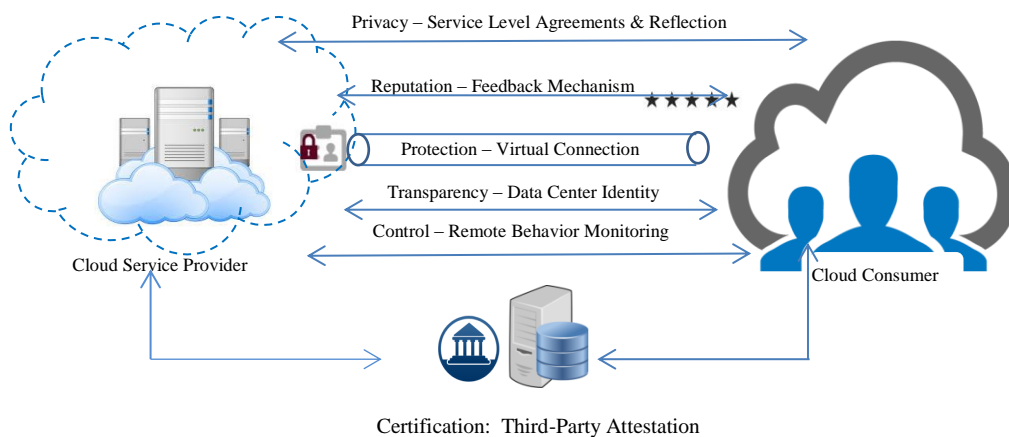


Figure 1. Trust Establishment between Service Provider and Consumer

Trust is a property that can be established through formal modeling of a system. Trust in a digital environment must be designed rigorously using advanced methodology involving human relationships, behavior changes, and a reputation index. Trust is an assessment that a person, organization, or object can be counted on to perform according to a given set of standards. In particular, a system is trusted only if its users trust it. Trust itself is an assessment made by users, based on how well the observed behavior of a system meets its own standards [13].

The primary goal of the work is to enhance security in a cloud environment by implementing a suitable model that enables trust policies to mitigate vulnerabilities. This could well be achieved through trusted communication implemented through an attestation server. Here all metrics related to software services are maintained and compared with earlier states of successful deployment. Since the cloud delivery model is constructed on service-oriented architecture, we have multiple trust nodes for cloud consumers and cloud providers, channelized through cloud brokers.

3.1. Integrity Assessment

Initial trust for a service is null for the first run because reputation, feedback, and behavior are not measured. After an evaluation process to check the integrity of a system, its process and data demonstrate progress. *System integrity* is a binary process, indicating whether or not the system has a trustworthy execution. This evaluation process is carried out by a trusted platform module, which provides a root of trust, whose computation and memory are tamperproof. In our work, we assume that all machines taking part in the process of trust evaluation have a TPM crypto processor embedded inside their motherboards. *Process integrity* depends on the codes executed during instantiation and should not be compromised. A modified code will yield malicious behavior on the part of the system, leading to mistrusted activity. A cryptographic process with a hash function can prevent intrusion. *Data integrity* can be ensured by checking the communication process. Hence, the stored measurement log, attestation key, and knowledge of fingerprinting are essential entities for assessment in our work.

4. Adaptation Algorithm

The adaptation algorithm involves taking appropriate action if and when, metrics deviate from the path of non-compliance [14]. The plan of the algorithm is to take necessary alternatives associated with the adaptation of each service. As soon as a cloud consumer requests service from a provider, the latter cooperates with sundry vendors for the action necessary. Thereafter, a connection establishment is initiated to map logical-physical connectivity. A process of authorization and authentication is invoked for granting the user access. Once the user passes all security procedures, an Application Monitoring Engine (AME) is initiated for supporting the adaptation procedure. Once the service starts initializing, 5 major phases are involved:

1. Training
2. Change detection
3. Verification
4. Decision,
5. Adaptation

In the *training phase*, measurement attributes from metrics like the mobility of the resource, network speed, and hardware configurations are collected. This data set is sent to a system integrity check process through the AME. Prior to this, the cloud service provider would have dispatched the necessary check value - to be used as a threshold for clearing the integrity check - to the behavior analytics server. The Behavior Analytics and Attestation Server (BAAS), functions as a change detector, verifier, and certifier. The key function of the BAAS is to verify the user, initially with the well-known values accorded by the service provider. Further, it calculates the trust score based on the weightage factor provided by the algorithm governing the system.

In the *change detection phase*, metrics being monitored by the AME are apprehended for exceptions. Checks taken for consideration include the following:

1. Failure of data transfer
2. TTL failure
3. IP check failure
4. User login failure
5. Boot process corrupted
6. Insufficient memory
7. Insufficient network bandwidth

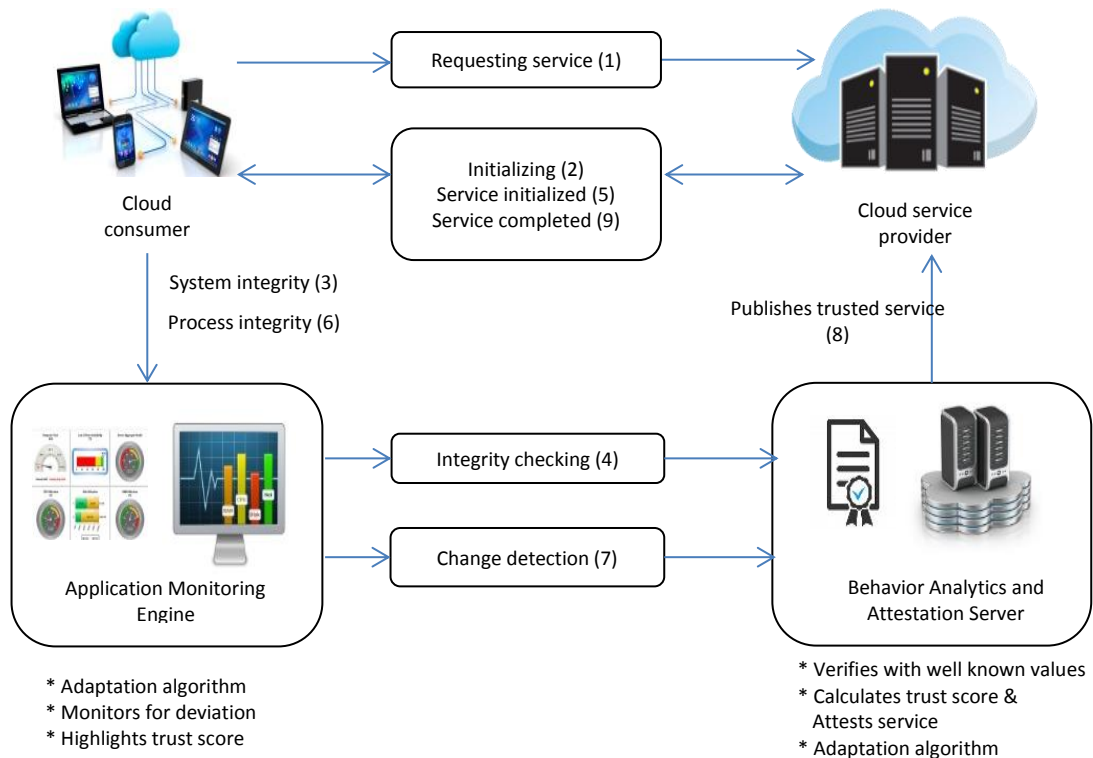


Figure 2. Trusted Cloud Architecture with BAAS and AME

In the *verification phase*, these values are cross-examined with data available in the BAAS. Similarly, we can provide a threshold to identify possible cases such as the following:

1. Maximum retransmission
2. Maximum delay
3. Time-out

The verification phase identifies the current state of the system, validates accurately what contributes to the events in question, and ascertains whether the changes caused are proven to be true. In this way, the process checks frequently for fresh values as contributory evidence. For example, in-case of failures with the TTL or data transfer, it checks for failure in the machine boot process to find whether the server has crashed.

Once it verifies that the changes have indeed happened and is confirmed, the *decision phase* is invoked to decide on the possible outcome of the system. It can either report to the adaptation phase to restart the system or increase bandwidth, if not then restart the application from the beginning. The decision module is the key to the evaluation of the trust score for the application, based on the earlier run.

```

procedure Adaptation_Algorithm
    while (SaaS is running) do{
        initiate AME
        monitor system_integrity
        Event ← Change_detection_AME ()
        if Event ≠ Normal
            invoke verification_analysis (Event_type, SaaS) and
                adaptation_plan (Event_cause, SaaS)
        else send value_system_integrity → BAAS
    } end adaptation_algorithm
procedure Change_detection_AME()
    if SaaS_initialization > maximum_delay
        Event_type = intercommunication_delay
    else if TTL failed
        Event_type = server_not_found
    .
    .
    else
        Event_type = Normal
    end if
    return (Event_type)

procedure Verification_analysis (Event_type, SaaS)
    case Event_type = = intercommunication_delay
        verify delay at consumer side
        if measure_delay > maximum_delay
            check if delay is caused by heavy load at cloud service provider
                Event_cause = heavy_load_at_CSP
            end if
        return (Event_cause)

procedure Adaptation_plan (Event_cause, SaaS)
    case Event_cause = heavy_load_at_CSP
        if additional VM required
            report CSP to assign new VM thru BAAS
        if partial recovery is possible
            resume from task checkpoint state
        else
            start the SaaS from the beginning
        end if
    end
    
```

5. Implementation

The implementation of the work is done using OpenStack, a platform for deploying cloud computing with components controlling hardware pools of processing, storage, and network resources. In terms of cloud computing terminology, OpenStack is an open source platform that lets you build Infrastructure as a Service (IaaS), a cloud that runs on commodity hardware. We have deployed OpenStack on two systems with Ubuntu as the host operating system. The first one acts as the controller node and the other as the compute node. The controller holds the Apache web server stack and runs the control service, message queue, and identity services. The controller runs the SaaS application developed in JavaScript and PHP, with an extension running in the Google Chrome browser. The extension acts as a plugin, whereby if the application hosted in the floating IP number 10.0.0.4 gets invoked, the plugin automatically starts assessing the cloud

consumer. This additionally monitors the application’s behavior. The compute node holds the MySQL database that holds the measurement values being monitored for the SaaS application. As depicted in Figure 3, the cloud implementation stack runs within a closed network on an open-sourced environment.

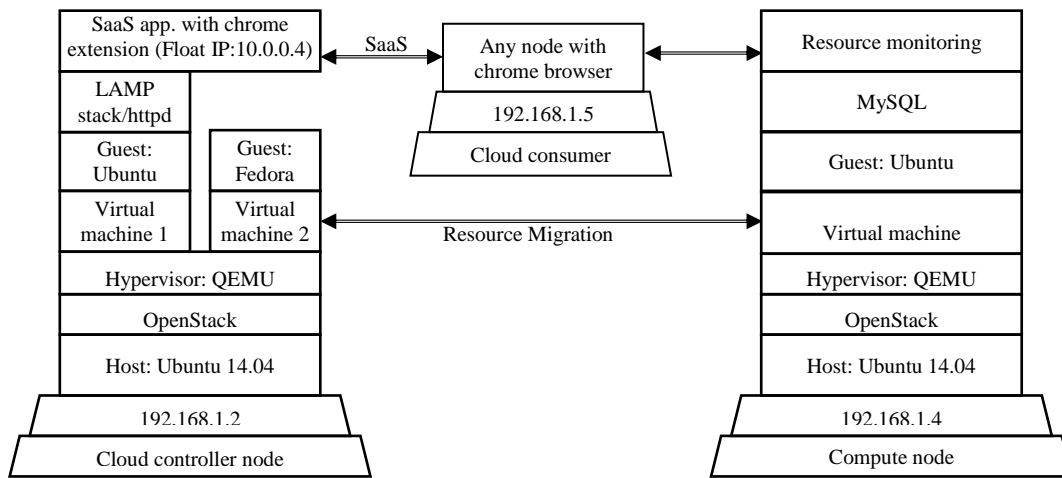


Figure 3. Implementation Stack of a Cloud Environment

The migration of resources from the compute node can be visually seen in the cloud controllers through the OpenStack Dashboard interface, as shown in Figure 4.

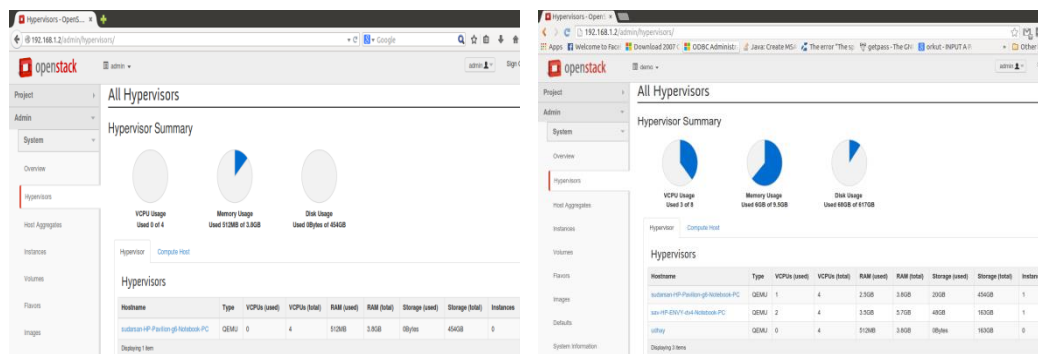


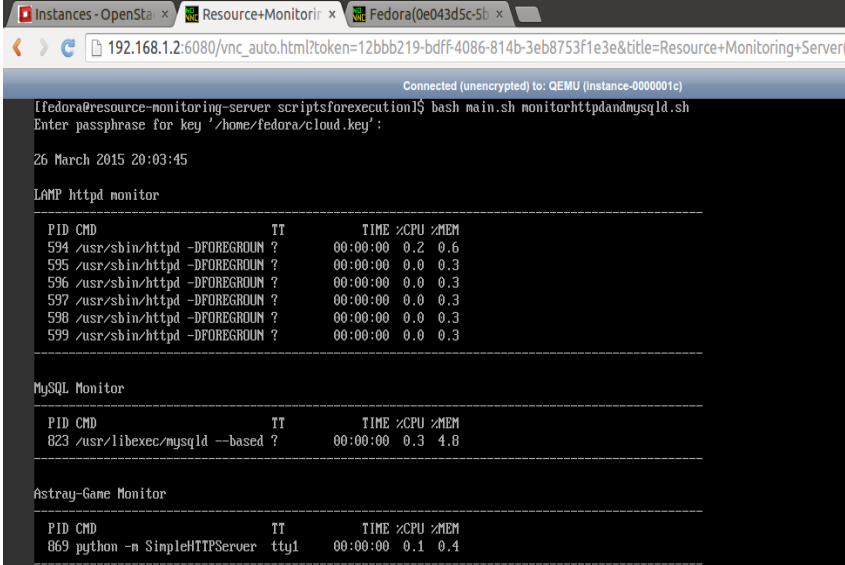
Figure 4. Hypervisor Summary of OpenStack, Before and After Migration

It can be seen from Table 1 that resources such as a virtual CPU, memory, and disk storage initially, hold the controller’s usage statistics. Once the compute node is connected, its resources are then added to the maximum extent of an 8 virtual CPU and 9.5 GB of memory space. After resource migration, the LAMP stack is set with an httpd start, enabling the SaaS application to run in a floating IP.

Table 1. Before and After Resource Migration

Major resources	VCPU		RAM		Disk usage	
	Usage	Total	Usage	Total	Usage	Total
Before migration	0	4	512MB	3.8GB	0 Byte	68GB
After migration	3	8	6GB	9.5GB	454GB	617GB

A shell script for monitoring the httpd, MySQL and the application is executed, the screen shot for which is shown in Figure 5. A simple image editing application is loaded and its complexity is analyzed.



```
[fedora@resource-monitoring-server scriptsforexecution]$ bash main.sh monitorhttpdandmysqld.sh
Enter passphrase for key '/home/fedora/cloud.key':

26 March 2015 20:03:45

LAMP httpd monitor
-----
PID CMD                IT          TIME %CPU %MEM
594 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.2 0.6
595 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.0 0.3
596 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.0 0.3
597 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.0 0.3
598 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.0 0.3
599 /usr/sbin/httpd -DFOREGROUND ? 00:00:00 0.0 0.3
-----

MySQL Monitor
-----
PID CMD                IT          TIME %CPU %MEM
823 /usr/libexec/mysqld --based ? 00:00:00 0.3 4.8
-----

Astray-Game Monitor
-----
PID CMD                IT          TIME %CPU %MEM
869 python -n SimpleHTTPServer tty1 00:00:00 0.1 0.4
-----
```

Figure 5. Monitoring of httpd, MySQL, and SaaS Applications

It can thus be seen that a virtual machine can effectively monitor the software's resources, being offered as a service by another VM. This VM can be deployed in a third-party server to establish reliable resource monitoring with attestation. It is in this way, that our model has implemented a good and trustworthy architecture that is both naturally dynamic and adaptable.

6. Conclusion

The dependability and integrity of cloud environment should improve the consumer's confidence and help new users to utilize applications and assure consumers of its commitment towards trustworthiness. However, certain issues and apprehensions encompassing cloud infrastructure have induced caution and reluctance to adopt it. Hence, it is necessary to monitor services through trust models and policies. Our proposed trust model assures confidence through behavior analysis using AMS, and attestation using BAAS. The model provides an essential framework for future cloud access, that can be trusted based on its dynamic nature and real-time implementation. The OpenStack cloud implementation proves to be more effective than other models, as the adaptation algorithm effectively plans the course of action if any changes in behavior are found. Thus, the very idea of establishing relational behavior in a digital world of services would certainly make the system behave as expected, and this expectation will lead to a trustworthy cloud computing.

References

- [1] "Code of practice for information security controls based on ISO/IEC 27002 for cloud services", ISO/IEC 27017:2015 Information technology -- Security techniques (2015).
- [2] W. Hussain, F Khadeer Hussain and O Khadeer, "Maintaining Trust in Cloud Computing through SLA Monitoring", Neural Information Processing, Lecture Notes in Computer Science, Springer International Publishing, vol. 8836, (2014), pp.690-697.
- [3] "Trusted Multi-Tenant Infrastructure Reference Framework", from Trusted Computing Group, (2013).

- [4] L. Zhenpeng, W. Xu, L.Yifa, G. Ding and Z. Xianchao, "Client-Oriented Remote Attestation Model in Cloud Environment", *International Journal of Security and Its Applications*, vol.9, no.10 (2015), pp.395-404.
- [5] D. Xian-Rui, Mu and Wei, "Research on Application of Hierarchical Trust Mechanism in Internet of Things", *International Journal of Security and Its Applications*, vol. 9, no. 6, (2015), pp. 101-114.
- [6] U. Shanmugam, Chandrasekaran, L Tamilselvan and F Ahmed, "An Adaptive Trust Model for Software Services in a Hybrid Cloud Environment", *Recent Researches in Computer Science, Proceedings of the 15th WSEAS International Conference on Computers*, September, (2011), pp. 497-502.
- [7] L. Lilien and B. Bhargava, "Privacy and Trust in Online Interactions", in *Online Consumer Protection: Theories of Human Relativism*, Idea Group, (2009), pp. 85-122.
- [8] Y. Prayudi1 and T. K. Priyambodo, "Secure and Trusted Environment as a Strategy to Maintain the Integrity and Authenticity of Digital Evidence", *International Journal of Security and Its Applications*, vol. 9, no. 6 (2015), pp. 299-314.
- [9] U. Shanmugam, L Tamilselvan, U Nandhini and Dhinakaran, "Attestation for Trusted Computing to Assure Security in Cloud Deployment Services", in *International Journal of Information and Electronics Engineering*, vol. 2, no. 4 (2012), pp. 644-648.
- [10] Z. Yan and S. Holtmanns, "Trust Modeling and Management: From Social Trust to Digital Trust", *Computer Security, Privacy and Politics: Challenges and Solutions*, IGI Global, (2007).
- [11] S. Boon, and J. Holmes, , "The dynamics of interpersonal trust: Resolving uncertainty in the face of risk", In R. Hinde and J. Groebel (Eds.). *Cooperation and Prosocial Behavior*. Cambridge University Press, Cambridge, UK, (1991), pp. 190-211.
- [12] K. M. Khan and Q Malluhi, "Establishing Trust in Cloud Computing", *IT Pro*, IEEE Computer Society, October, (2010), pp. 20-26.
- [13] D.E. Denning, "A new paradigm for trusted systems", in *IEEE New Paradigms Workshop*, (1993).
- [14] D. Kim and S. Hariri, "Adaptive Distributed Virtual Computing Environment (ADViCE)", *Virtual Computing*, In the Springer International Series in Engineering and Computer Science, vol. 633, (2001), pp. 35-51.

Authors



Udhayakumar Shanmugam, He received his M.E. degree in Multimedia Technology from the School of Computer Science and Engineering, College of Engineering, Anna University. He is currently doing his Ph.D. in the field of cloud security at B.S.Abdur Rahman Univeristy. His main research interests include Cloud Computing, Mobile Computing, Web Technology and Internet of Things.



Latha Tamilselvan, She is a Ph.D., Professor and Head of the Department of Information Technology, B.S. Abdur Rahman University. She has obtained her Ph.D. in Computer Science and Engineering from Anna University in 2008. Her research interest includes Cloud Computing, Mobile Ad hoc Network and Network Security